IN THE CLAIMS

Please amend the claims as follows:

1. (Original) A method for sharing data within a distributed computing system having a plurality of nodes, wherein each node includes a processor and memory, the method comprising:

distributing an application across two or more nodes, including a first and a second node, wherein distributing the application includes creating application processes, associating the application processes with a group and distributing the application processes to the first and second nodes;

defining a data object in memory on the first node, wherein defining a data object includes allocating memory for the data object;

assigning a name to the data object;

mapping the name to a data object handle; and

accessing data within the data object through references to the data object handle.

2. (Original) The method according to claim 1, wherein defining a data object further includes setting a data sharing policy for the data object.

3. (Original) The method according to claim 1, wherein accessing data includes querying a mapping service process with the data object name in order to retrieve the data object handle associated with the data object name.

4. (Original) The method according to claim 1, wherein accessing data includes determining ownership of the data object and requesting transfer of the ownership of the data object to a process on the second node.

5. (Original) The method according to claim 1, wherein associating the application processes with an application group includes establishing a group service process on one or more nodes, wherein each group service process tracks membership of processes in groups.

6. (Original) The method according to claim 5, wherein each group service process tracks formation of an application group, joining of processes to the application group and exit of application processes from the application group.

7. (Original) The method according to claim 5, wherein the group service process differentiates between accesses to a data object stored within memory of the node on which the group service process executes and accesses to a data object stored within memory of another node.

8. (Original) The method according to claim 5, wherein the group service process synchronizes access to and sharing of data structures between application processes on different nodes.

9. (Original) The method according to claim 5, wherein accessing data includes querying a mapping service process with the data object name in order to retrieve the data object handle associated with the data object name.

10. (Original) The method according to claim 5, wherein accessing data includes determining ownership of the data object and requesting transfer of the ownership of the data object to a process on the second node.

11. (Original) The method according to claim 1, wherein associating the application processes with an application group includes:

registering the application as a group; and

establishing a group service process on one of the nodes, wherein the group service process tracks membership of processes in groups.

12. (Original) The method according to claim 11, wherein accessing data within the data object includes obtaining a coherency state for the data object, wherein obtaining a coherency state includes accessing the group service process via the data object handle.

13. (Original) The method according to claim 1, wherein associating the processes with an application group includes establishing a group service process on each of the nodes, wherein the group service process on each node tracks membership in groups of processes local to its respective node.

14. (Original) The method according to claim 13, wherein each group service process tracks formation of a process group, joining of local processes to the process group and exit of local processes from the process group.

15. (Original) The method according to claim 13, wherein the group service process executing on the first node differentiates between accesses to a data object stored within memory of the first node and accesses to a data object stored within memory of another node.

16. (Original) The method according to claim 13, wherein each data object is owned by a owner process and wherein the group service process of the node where the owner process executes synchronizes access to and sharing of its respective data object between processes on different nodes.

17. (Original) The method according to claim 1, wherein mapping the name to a data object handle includes establishing a mapping service process on one of the nodes.

18. (Original) The method according to claim 13, wherein accessing data within the data object includes obtaining a coherency state for the data object wherein obtaining a coherency state includes accessing, via the data object handle, the group service process local to the node of the process seeking access to the data object.

19. (Original) The method according to claim 13, wherein accessing data includes querying a mapping service process with the data object name in order to retrieve the data object handle associated with the data object name.

20. (Original) The method according to claim 13, wherein accessing data includes querying a mapping service process with the data object name in order to retrieve the data object handle associated with the data object name and querying one or more group service processes with the data object handle in order to access the data object.

21. (Original) The method according to claim 1, wherein accessing data within the data object includes obtaining a coherency state for the data object via the data object handle.

22. (Original) The method according to claim 1, wherein mapping the name to a data object handle includes exporting the data object handle to a local group service process.

23. (Original) The method according to claim 22, wherein exporting includes accumulating two or more data object handles and exporting the two or more data object handles to the local group service process with a single export call, wherein the two or more data object handles are grouped in a vector within the export call.

24. (Original) The method according to claim 1, wherein allocating memory for the data object includes dividing a data structure into a number of data objects, wherein the number is a function of the type of application being executed.

25. (Original) The method according to claim 24, wherein mapping the name to a data object handle includes exporting the data object handle associated with each data object to a local group service process.

26. (Original) The method according to claim 25, wherein each data object has a data object size that is the smallest unit of update of the data object.

27. (Original) The method according to claim 1, wherein associating the processes with a group includes registering the application as a group and joining the processes to the group.

28. (Original) An article comprising a computer readable medium having instructions thereon, wherein the instructions, when executed in a computer, create a system for executing the method of claim 1.

29. (Original) An article comprising a computer readable medium having instructions thereon, wherein the instructions, when executed in a computer, create a system for executing the method of claim 6.

30. (Original) An article comprising a computer readable medium having instructions thereon, wherein the instructions, when executed in a computer, create a system for executing the method of claim 11.

31. (Original) An article comprising a computer readable medium having instructions thereon, wherein the instructions, when executed in a computer, create a system for executing the method of claim 13.

32. (Original) A method for ensuring consistency in shared data within a distributed computing system having a plurality of nodes, wherein each node includes a processor and memory, the method comprising:

distributing an application across two or more nodes, including a first node, wherein distributing the application includes creating processes, associating the processes with a group and distributing the processes to the two or more nodes;

establishing a data object in memory of the first node, wherein establishing a data object includes allocating memory for the data object and assigning a name to the data object;

making the data object available to processes associated with the group, wherein making the data object available to processes associated with the group includes:

setting a data sharing policy for the data object;

mapping the data object name to a data object handle; and

assigning a data length to the data object representative of the memory allocated to the data object;

returning the data object handle to a process associated with the group in response to a request from the process based on the data object name; and

accessing data stored in the data object through references to the data object handle.

33. (Original) The method according to claim 32, wherein associating the processes with a group includes:

registering the application as a group;

establishing a group service process on one or more of the nodes, wherein the group service process tracks membership of processes in groups; and

establishing a mapping process on one of the nodes, wherein the mapping process provides the data object handle and the data length to processes associated with the group in response to a request based on the data object name.

34. (Original) An article comprising a computer readable medium having instructions thereon, wherein the instructions, when executed in a computer, create a system for executing the method of claim 32.

35. (Original) A method for ensuring consistency in shared data within a distributed computing system having a plurality of nodes, wherein each node includes a processor and memory, the method comprising:

distributing an application across two or more nodes, including a first node, wherein distributing the application includes creating processes associated with the application and distributing the processes to the two or more nodes;

establishing a data object in memory on the first node, wherein establishing a data object includes allocating memory for the data object and assigning a data object name to the data object;

mapping the data object name to a data object handle, wherein mapping includes exporting the data object handle to a local service process executing on the first node; and

accessing the data object on the first node via the data object handle, wherein accessing the data object includes querying the local service process with the data object handle.

36. (Original) The method according to claim 35, wherein exporting includes accumulating two or more data object handles and exporting the two or more data object handles to the local group service process with a single export call, wherein the two or more data object handles are grouped in a vector within the export call.

37. (Original) An article comprising a computer readable medium having instructions thereon, wherein the instructions, when executed in a computer, create a system for executing the method of claim 35.

38. (Original) A method for ensuring consistency in shared data within a distributed computing system having a plurality of nodes, wherein each node includes a processor and memory, the method comprising:

distributing a first application across two or more nodes, including a first node and a second node, wherein distributing the application includes creating processes, associating the processes with a first process group and distributing the processes to the two or more nodes;

establishing a group service process, wherein the group service process tracks processes joining and leaving the first process group;

establishing a data object in memory on the first node, wherein establishing a data object includes assigning a name to the data object, setting a data sharing policy for the data object, and determining a data length associated with the data object;

assigning a data object handle to the data object;

returning the data object handle to the second node in response to a request based on the data object name; and

accessing the data object through references to the data object handle.

39. (Original) The method according to claim 38, wherein accessing the data object includes obtaining information on ownership of the data object and a coherency state for data within the data object.

40. (Original) The method according to claim 38, wherein associating the processes with a first process group includes:

registering the first application as a group; and

establishing a group service process on one of the nodes, wherein the group service process tracks membership of processes in groups; and

wherein assigning a data object handle to the data object includes establishing a mapping process on one of the nodes, wherein the mapping process returns a unique data object handle for each data object.

41. (Original) An article comprising a computer readable medium having instructions thereon, wherein the instructions, when executed in a computer, create a system for executing the method of claim 38.

42. (Currently Amended) A method for ensuring consistency in shared data within a distributed computing system ~~shaving~~ having a plurality of nodes, wherein each node includes a processor and memory, the method comprising:

distributing a first application across two or more nodes, including a first node, wherein distributing the application includes creating processes, associating the processes with a first process group and distributing the processes to the two or more nodes;

establishing a local group service process on each node, wherein each local group service process tracks processes joining and leaving the first process group from that node;

establishing a data object in memory on the first node, wherein establishing a data object includes allocating memory for the data object, assigning a name to the data object and setting a data sharing policy for the data object;

assigning a data object handle to the data object, wherein the data object handle includes information regarding the data sharing policy and a memory size label representative of memory allocated for the data object, wherein assigning includes recording the data object name and the data object handle so that the data object handle can subsequently be determined from the data object name; and

exporting the data object handle to the local group service process executing on the first node.

43. (Original) The method according to claim 42, wherein associating the processes with a first process group includes:

registering the first application as a group;

establishing a group service process on one of the nodes, wherein the group service process tracks membership of processes in groups; and

establishing a mapping process on one of the nodes, wherein the mapping process provides a unique data object handle to be assigned to the data object.

44. (Original) An article comprising a computer readable medium having instructions thereon, wherein the instructions, when executed in a computer, create a system for executing the method of claim 42.

45. (Original) A distributed computing system, comprising:

a plurality of intercommunicating nodes, wherein each node includes a processor and memory;

means for distributing an application as processes running on two or more nodes, wherein the means for distributing includes a group service process running on one of the nodes, wherein the group service process tracks processes joining and leaving a process group associated with the application; and means for exporting a data object associated with the application to processes within the process group, wherein the means for exporting includes means for assigning a data object handle to the data object, wherein the data object handle uniquely identifies the data object;

wherein the data object handle is used to obtain a coherency state of the data object for processes within the process group.

46. (Original) The system according to claim 45, wherein the system further includes means for accessing data within the data object, wherein the means for accessing data includes means for obtaining a coherency state for the data object.

47. (Original) The system according to claim 45, wherein the group service process tracks group memberships of processes local to its respective node.

48. (Original) The system according to claim 47, wherein each node includes means for multicasting to the other nodes a request to identify who owns the data object.

49. (Original) A distributed computing system, comprising:

a plurality of intercommunicating nodes, wherein each node includes a processor and memory;

means for distributing an application as application processes running on two or more nodes, wherein the means for distributing includes a group service process running on two or more of the nodes, wherein the group service processes track application processes joining and leaving a process group associated with the application; and

means for making a data object associated with the application available to processes within the process group, wherein the means for making a data object associated with the application available to processes within the process group includes means for assigning a data object handle to the data object and means for writing the data object handle to one of the group service processes, wherein the data object handle uniquely identifies the data object;

wherein the application processes use the data object handle to access data within the data object.

50. (Original) The system according to claim 49, wherein the system further includes means for obtaining a coherency state associated with the data object handle.

51. (Original) The system according to claim 49, wherein the group service process tracks group memberships of processes local to its respective node.

52. (Original) The system according to claim 51, wherein each node includes means for multicasting to the other nodes a request to identify who owns the data object.

53. (Original) A computing system for executing a distributed application, the computing system comprising:

a plurality of intercommunicating nodes, wherein each node includes a processor and memory, wherein one of the memories includes a data object, wherein the data object has a data object name;

a mapping process executing on one of the plurality of nodes, wherein the mapping process maps the data object name to a unique data object handle, wherein the mapping process returns the data object handle in response to a request from a process based on the data object name; and

a group service process operating on each node, wherein the group service process tracks processes local to its node as they join and leave a process group associated with the distributed application and wherein the group service process responds to requests, based on the data object handle, for access to the data object associated with the data object handle.

54. (Original) The computing system of claim 53, wherein the data object handle is used to obtain a coherency state of the data object for processes within the process group.

55. (Original) The computing system according to claim 53, wherein the system further includes means for accessing data within the data object, wherein the means for accessing data includes means for obtaining a coherency state for the data object.

56. (Original) The system according to claim 53, wherein each node includes means for multicasting to the other nodes a request to identify who owns the data object.

57. (Original) A method for ensuring consistency in shared data within a distributed computing system having a plurality of nodes, wherein each node includes a processor and memory, the method comprising:

distributing an application across two or more nodes, including a first node, wherein distributing the application includes creating processes, associating the processes with a group and distributing the processes to the two or more nodes;

establishing a first and a second data object in memory on the first node, wherein establishing each data object includes assigning a name to each data object and allocating memory for each data object;

setting a data sharing policy for each data object;

mapping the first and second data object names to first and second data object handles, respectively;

exporting the first and second data objects;

returning the first data object handle in response to a request based on the first data object name, wherein returning the first data object handle includes returning information regarding data sharing policy and data object size for the first data object; and

accessing the first data object using the first data object handle.

58. (Original) The method according to claim 57, wherein accessing the first data object using the first data object handle includes obtaining a coherency state for the first data object through a reference to the first data object handle.

59. (Original) The method according to claim 57, wherein exporting the first and second data objects includes calling an API having a data object handle vector and a count, wherein the API writes the data object handles in the data object handle vector to a local group service process.

60. (Original) A router, comprising:

a plurality of router nodes, wherein each router node communicates with the other router nodes and wherein each router node includes a route processor and memory, wherein the memory includes a data object and wherein the data object has a name;

a mapping process executing on one of the plurality of nodes, wherein the mapping process maps the data object name to a unique data object handle, wherein the mapping process returns the data object handle in response to a request from a process based on the data object name; and

a group service process operating on each node, wherein the group service process tracks processes local to its node as they join and leave a process group associated with the distributed application and wherein the group service process responds to requests, based on the data object handle, for access to the data object associated with the data object handle.

61. (Currently Amended) The ~~system~~ router according to claim 60, wherein the data object handle is used to obtain a coherency state of the data object for processes within the process group.

62. (Currently Amended) The ~~system~~ router according to claim 60, wherein the router further includes means for accessing data within the data object, wherein the means for accessing data includes means for obtaining a coherency state associated with the data object.

63. (Currently Amended) The ~~system~~ router according to claim 60, wherein each node includes means for multicasting to the other nodes a request to identify who owns the data object.